

LINUX EXPERT

FAQ

Q Are there any useful key combinations in Linux that allow you to do things such as display the running process list or switch between desktops quickly?

A Linux supports multiple terminals, which allow you to log into the system multiple times and switch between sessions quickly. You can move between them by holding down the Ctrl key and pressing F1 for the first session, F2 for the second and so on.

You can bring up the running process list – as you can in Windows XP by pressing Ctrl-Shift-Esc – by pressing Ctrl-Esc. This will allow you to deal with runaway processes, including those that refuse to stop or exit on command.

You can even access a number of Linux command lines by pressing Ctrl-Alt-F1 through to Ctrl-Alt-F6. These will present you with a login prompt, and are also useful if you're running without a graphical user interface as they give you access to six terminals in parallel. To get back to your graphical desktop, simply press Ctrl-Alt-F7.

Jon Thompson
UNIX/Linux management
and security consultant



linux@computershopper.co.uk

Boost your web speed

Installing a web proxy can speed up your internet connection and let you control your users' online activity. Jon Thompson shows you how Squid can make you master of your domain

Those who share an internet connection between office workers or members of the family are often asked two interesting questions: can you restrict access to certain sites, and is it possible to speed things up? While speed problems usually lie outside your local network, your users or family will often be quick to blame you. Luckily, in this month's *Linux Expert* we show you a solution to both these problems that's simple to set up and free to use.

What you need to do is install a web proxy. Squid is a fully featured, open-source caching web proxy system that sits on your network between the desktop PCs and the internet connection. It supports HTTP, HTTPS and FTP, making it ideal for both home and business use.

In general terms, a proxy is a server that stands between the source and destination of a connection. Without a proxy, a client program can communicate directly with a server. The proxy usually intercepts the client's request for communication and connects to the server on its behalf. It receives instructions from the client and passes them on to the server, while feeding back the server's responses to the client.

Because the proxy prevents the client interacting directly with the server, it can be used to monitor the client's behaviour and control its requests and the data it can receive. A web proxy works in this way, visiting websites and fetching pages and other information on behalf of the computers on a network. The proxy can be told to block access to certain sites and it can also cache the information it retrieves from the web. This means it can serve this content locally rather than having to fetch it fresh from the web every time. This allows people on your network to access information that has already been viewed faster.

A simple adjustment to web browsers such as Firefox and Microsoft's Internet Explorer will allow them to access Squid as if they were accessing the internet directly, and your users will be none the wiser unless you tell them. They might even notice that the internet appears to have started working more quickly.

SETTING UP SQUID

Squid needs access to plenty of RAM to function efficiently. It runs as a single process and, for performance reasons, it keeps a RAM-based index to all the cached objects that comprise the pages it has served recently. The objects themselves are stored in a disk-based cache, with popular pages usually held automatically by the kernel in fast disk buffers, thereby lessening retrieval times. The reason for it needing lots of RAM isn't difficult to see.

Having too little memory means that there will be delays when the kernel swaps the Squid process or parts of its index from disk swap space. This will slow down web access noticeably, making things worse than if the users were accessing websites directly. At least 256MB of RAM will prevent this problem when setting up a small, dedicated Squid proxy for a home network. For larger companies that are concerned about bandwidth usage and access times, 512MB or 1GB is preferable.

As Squid does not place heavy demands on the processor, you can install it on a fairly old PC. The documentation recommends that you use disk drives

with very fast random seek times, but this becomes an advantage only when you have large numbers of users all surfing at once, which puts Squid under increased pressure to perform as efficiently as possible.

The Squid system should have plenty of disk space to store its log files, though. These hold detailed information and can grow very large when used on a busy system. As this is a server that should be accessed by computers on the internal network only, it should also be installed on the internal network. Because it won't be publicly available to other internet users, you don't need to spend as long securing it or install it in a demilitarised zone, where you'd normally connect your public web and mail servers.

STATIC SPACE

You also need to assign a static IP address to the server so that the web browsers on your PCs can find it without having to resolve its name using DNS first. You can set a static IP address using your distribution's graphical configuration tool and turn off DHCP at the same time. Using the command line is a little more complex, but it's not very hard. As root, issue the ifconfig command to display the MAC address of the network interface card, then edit the file called /etc/sysconfig/network/ifcfg-eth-id-<MAC address>. Replace the <MAC address> part with the series of characters that ifconfig displayed earlier.

The line that tells the network interface to broadcast a request for a DHCP lease is:

```
B00TPROT0='dhcp'
```

Change 'dhcp' to 'static'. Then assign an IP address to the network interface card and check it using the following commands, replacing 192.168.0.1 with the IP address you configured:

```
# ifconfig eth0 192.168.0.1 netmask
255.255.255.0 up#
# ifconfig eth0#
eth0 Link encap: Ethernet Hwaddr 00:0A:
E6:BD:C2:89
inet addr:192.168.0.1 Bcast:
192.168.0.255 Mask:255.255.255.0
inet6 addr: fe80::20a:e6ff:febd:c289/64
Scope:Link
UP BROADCAST NOTRAILERS RUNNING
MULTICAST MTU:1500 Metric:1
RX packets:8 errors:0 dropped:0
overruns:0 frame:0
TX packets:28 errors:0 dropped:0
overruns:0 carrier:0
Collisions:0 txqueuelen:1000
RX bytes:1559 (1.5Kb) TX bytes 3926
(3.8 Kb)
Interrupt:5 Base address:0xcfc00
```

Pinging the system from another computer should also verify that the card is up and listening on, in this case, IP address 192.168.0.1:

```
# ping 192.168.0.1#
```



Squid is included with most mainstream Linux distributions and is incredibly simple to install with a graphical installation tool such as YaST or Anaconda. For those of you who are unlucky enough not to have Squid included in your distribution, it's easy to download, unpack, configure and build, but you need to ensure you have your C compiler installed first.

You can obtain the source files from `ftp://ftp.squid-cache.org/pub/squid-2/STABLE`. The file to download is `squid-2.6.STABLE9.tar.gz`. Save it to the `/usr/local` directory and unpack it into its own subdirectory with the following command:

```
# tar xvf squid-2.6.STABLE9.tar.gz
```

The installation process then follows a familiar path. You just have to enter the directory into which tar placed Squid's files and enter the following commands:

```
# ./configure
# make
# make install
```

TESTING TIMES

All you have to do to configure your browser to use Squid is change its proxy settings to make a call to port 3128 on the IP address of the Squid server. This is as opposed to contacting port 80 on a remote IP address, as it does by default. See the 'Configuring your browser' box on page 170 for more details.

We also need to configure Squid to allow access to the proxy service on that port. For testing purposes, we'll allow any host to connect. To do this, edit the file called `/etc/squid/squid.conf` and find the following line:

```
# http_access deny all
```

Change this to read:

```
# http_access allow all
```

Then you need to start the Squid proxy with the following command:

```
# /etc/init.d/squid start
Starting WWW-squid done
```

On a client computer that is configured to access Squid, you simply start the browser. The usual homepage should come up. Then stop Squid as follows:

```
# /etc/init.d/squid stop
Shutting down WWW-squid done
```

Try accessing a webpage again and you should see an access-denied message generated by Squid, as shown in the screengrab below. You can tell it's coming from Squid because the banner page will contain the name of the proxy machine and the version of Squid being run on it. In our screenshot, the machine is called `linux.site` and the Squid version is `2.5.STABLE10`. Start the proxy server and you should now be able to see the outside world again.

You need Squid to run all the time so ensure it starts automatically at system bootup as follows. If your Linux distribution includes the very useful `insserv` command then enter the following command:

```
# insserv /etc/init.d/squid
```

This command will also start Squid immediately. If you change your mind and don't want the Squid server to run at boot time, you can run the command again, this time using the `-r` switch to remove it from the list of programs that start automatically:

```
# insserv -r /etc/init.d/squid
```

This doesn't stop Squid until you reboot. Some distributions don't include the `insserv` command. These include Debian-based ones such as Ubuntu. You can add the following line to your `/etc/inittab` file to have the server start automatically:

```
sq:35:respawn:/etc/init.d/squid start
```

This line will start Squid at runlevels 3 and 5, which correspond to Linux booting up into text-based and graphical modes with full networking support.

It's also worth noting that you can customise the error pages that Squid uses to notify your users of various conditions. These error pages are simply HTML files stored in the directory called `/etc/squid/errors`. The defaults are basic but functional, and if you know HTML you can improve on them.

ACCESS ALL AREAS

Squid also logs access attempts, both to its service and to webpages on the internet. This can be useful in cases where you trust your

users, but suspect someone of accessing pages they shouldn't during work time. The log files are stored in `/var/log/squid`, and there are usually four files in this directory.

The one called `access.log` contains a complete audit trail of all traffic to and from the web browsers on your network. The `cache.log` file logs all activities Squid carries out on its cache file, including setting it up, validating it after a reboot and any housekeeping tasks.

The `rcsquid.log` file contains warnings and errors that are generated when starting up Squid and reading the configuration file. This is the first port of call if you have any problems when you start the proxy or get any unexpected error messages on the console. Finally, the `store.log` log file shows when objects in the cache are written to or removed from disk.

Once up and running, you can get a quick snapshot of the latest web activity through Squid by using the `tail` command:

```
# tail -n 20 access.log
```

The output is best viewed in a terminal that has been maximised so that it fills the screen. If you want a more dynamic output, you can use `tail`'s `-f` switch to display new lines of log data as they are recorded:

```
# tail -n 20 -f access.log
```

To exit the `tail` command, simply hit `Ctrl-C`.

CONFIGURING SQUID

Once Squid is up and running, you will want to configure it to suit your needs and those of your users. Squid's configuration options are known as tags in the documentation and configuration file.

If all your web traffic runs through the proxy, and the proxy fails, it stands to reason that you and your users lose the ability to browse the web. A failure might occur in the rare situation where the cache becomes corrupt and Squid can't continue. In this case, you will need to be notified very quickly so that you can restart the proxy. To do this, add your email address to the option called `cache_mgr`, which is stored in the `squid.conf` file. You will be notified should Squid fail.

By default, Squid will send an email to the webmaster. Uncomment the `cache_mgr` field and change it to your preferred account, which should preferably be one that you're always logged into so that you'll be notified as soon as there's any problem. This email address is, if edited to your own preference, appended to any error messages your users see if they are served one of Squid's error pages. It becomes part of the message that usually reads: "Your cache administrator is webmaster."

Allied to `cache_mgr` is the mail program that Squid uses to send mail. This is defined by the `'mail_program'` tag. This is `'mail'` by default, and the `'mail_program'` tag needs to be uncommented and edited if you have a different preference.

In some circumstances, perhaps due to firewall restrictions, it is not possible for Squid to use its default port of 3128. You can change this, and even add extra alternative ports, using the `http_port` tag. For example, to add port 8080 to ports that Squid listens to, add this line:

```
http_port 3128 8080
```



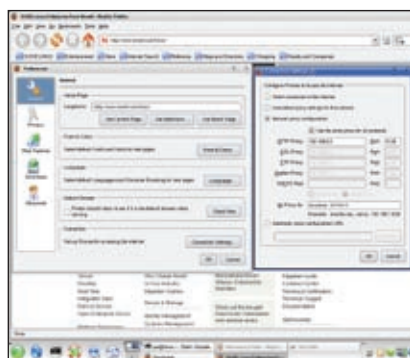
▲ The access-denied message generated by Squid when you are denied access to a webpage. The banner page contains the name of the proxy machine and the version of Squid that is being run on it

Ordering Squid Configuring your browser

For the entire network to access the web through your Squid proxy, you'll need to configure each PC's web browser. Different browsers handle proxy connections slightly differently, so we've included this handy guide for all the most popular ones included in Linux and Microsoft Windows.

FIREFOX 1.06 (LINUX)

This is the older version of Firefox shipped last summer with Linux distributions that contained a graphical desktop. Go to the Edit drop-down menu and select Preferences. Click on the General category on the left of the window that appears and press the Connection Settings button at the bottom. Click on the Manual proxy configuration radio button, then enter the IP address of the



▲ In Firefox 1.0.6 click General and press the Connection Settings button at the bottom

Squid proxy server, with 3128 as the port to use. Click OK on this window and OK on the parent window. Your web traffic should now go through the proxy.

FIREFOX 2 (WINDOWS)

For version 2 of Firefox, Mozilla changed and expanded the control you can exert over the way it handles proxy connections. Go to the Tools menu and click on Options. Click on the Advanced category and select the Network tab. Press the Connection Settings button to access the proxy details. Select the Manual Proxy Configuration radio button and enter the IP address of the Squid proxy server in the HTTP Proxy box. Enter 3128 for the port to use and tick the box called Use this proxy for all protocols. You then need to click the OK



▲ For Firefox 2 Mozilla changed the control you can exert over how it handles proxy connections

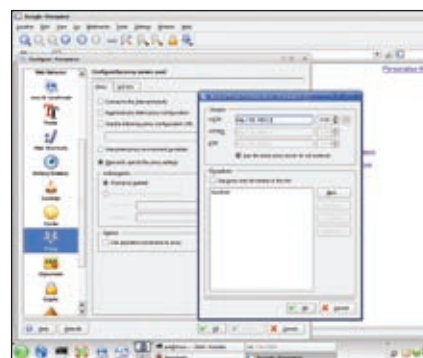
button on this window and the OK button on the parent window.

FIREFOX 2 (LINUX)

Configuring Firefox 2 on Linux is the same as for the Windows version except that, instead of starting with the Tools menu, you go to the Edit menu and select Preferences.

KONQUEROR (LINUX)

You can configure Konqueror to use a web proxy server by choosing the Settings menu and selecting Configure Konqueror. Scroll down the sidebar on the left of the subsequent window and select the Proxy icon. In the right-hand pane, click on the radio button marked Manually select the proxy settings, then press the associated Setup button.



▲ Configure Konqueror to use a web proxy server by choosing Settings and Configure Konqueror

Restart Squid and it will now listen on both ports 3128 and 8080.

It's important to remember that ports below 1024 are called privileged ports. This means that only processes owned by root can access them. If you've installed Squid to boot up at system startup time, the startup script will run as root. Once started, however, there's no need for Squid to continue to run as the super user and a good reason why it shouldn't: doing so poses a security risk. For security purposes, Squid switches to being owned by the user 'nobody'. You can change this, though, and specify a different user and group.

You do this by using the `cache_effective_user` and `cache_effective_group` tags. You can set these either to the numeric or symbolic UID and GID, as found in the file `/etc/passwd`.

You can set the hostname used in error pages to appear as something other than the machine's real hostname. This is useful in cases where you want users receiving error messages to see a friendlier server name. The tag for this is `visible_hostname`.

Another useful tag is `ftp_user`. Normally, when you use a public anonymous FTP server, you enter your email address as the password. It is just a polite way of doing things and means that, if there's a problem, the FTP site administrator knows who you are. When you perform an anonymous FTP transfer with your web browser and without the intervention of Squid, your browser will send a nonsense email address. The `ftp_user` tag contains the username to substitute, which should be your email address or that of an equally responsible person.

This is useful because, if anyone at your site abuses the FTP server, such as by trying to download too many things at once and tying up its bandwidth, the remote administrator can send you an email to ask you to sort out the problem.

LIVING IN DENIAL

Now that your users have a new, fast web-access system, let's use it to restrict their view of the internet. The most basic method of doing this is simply to use the following tag in `squid.conf`, which blocks access to all websites:

```
http_access deny all
```

You can use 'allow' instead of 'deny' to open the floodgates to all requests. In a realistic system you will have a number of `http_access` tags, each specifying different rules. The tag in the list should be 'http_access deny all', which acts as a catch-all for requests that have not been addressed by the previous rules.

It is possible to allow or deny access to the web based on a wide range of criteria, such as time of day or the MAC address of a machine's network interface card. Squid uses access control lists (ACLs) to define these rules.

A Squid ACL consists of several parts. First, you define the ACL itself. This has a user-defined name, a type and a string defining a pattern. The type tells Squid how to interpret the pattern. We then apply the ACL to an action, such as `http_access`. Here's an example of limiting access to a single IP address, setting up the ACL first and then applying it to the `http_access` action:

```
acl only_me src 192.168.0.2
http_access allow only_me
```

The ACL's name is `only_me`. The type is `src`, meaning that the pattern is a local source address. The pattern, `192.168.0.2`, is the IP address of my computer. The second line tells Squid what to do with the ACL. In this case, we want it to allow web access to the computer that has the IP address `192.168.0.2`.

The type parameter has a number of very useful values that we can use to exert precise control over the proxy. Here's a slightly more complex use of an ACL that will limit access to certain times of the day and deny it at all other times:

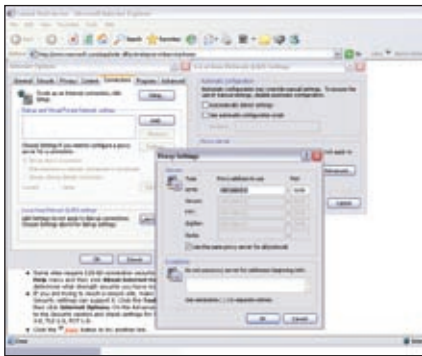
```
acl ip_acl src 192.168.0.2
acl time_acl time M T W H F 9:00-17:00
http_access allow ip_acl time_acl
http_access deny all
```

There are two ACLs. The 'ip_acl' ACL specified the address allowed to have access, and 'time_acl' ACL specifies the times this system will be granted access. The ACLs act as variables in a program and are used in the first `http_access` line. If the incoming connection request satisfies these criteria, Squid allows access and moves on to process the next incoming request. If not, it moves to the second `http_access` tag and denies access by default. You should always have a line like this as the final action in `squid.conf`. If you don't, access is granted to the proxy by default and you lose control of its use.

When a second window pops up, check the box marked 'Use the same proxy server for all protocols' and enter the IP address of the Squid server in the input line marked HTTP. Change the port from 8080 to 3128. Click the OK button on this window and the one on its parent window.

INTERNET EXPLORER 6 (WINDOWS)

From the Tools menu, you must select Internet Options. In the resulting window, select the Connections tab. Then select LAN settings. This causes a second window to appear. Click on the tickbox marked Use a proxy server for your LAN and press the Advanced button. This produces a third window. Click on the tickbox marked Use the same proxy server for all protocols, then enter the IP address of the

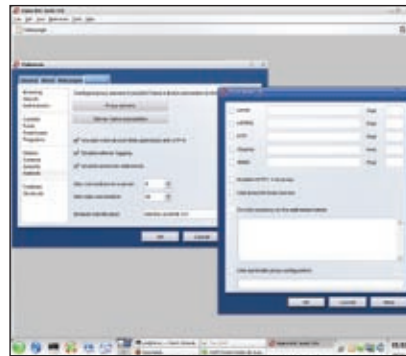


▲ In Internet Explorer 6, go to Tools, pick Internet Options and then select the Connections tab

Squid server and a port number of 3128. Click OK on all the windows and you should now be using the proxy to access the web.

OPERA (LINUX)

This often-overlooked web browser is an unsung gem, being fast, compact and secure. Configuring it for use with a proxy is a little more involved than for the others. Open the Tools menu and select Preferences. Select the Advanced tab and click on Network on the left-hand side of the window. Press the Proxy servers button to open a second window. Then click on the protocols (HTTP, HTTPS, FTP, Gopher and WAIS) and enter the IP address and port of the Squid server for each. Finally, press OK on this and the previous window.



▲ Configuring Opera for use with a proxy is a little more involved than for the others

This example shows that you can use more than one parameter with `http_access`. This tag accepts multiple ACLs and processes them in order. It's useful to read them as if they were an 'if...then' statement. So, if the incoming request satisfies the criteria set down in `ip_acl` and `time_acl`, then allow access.

This slightly more complex example restricts access to different IP addresses according to time ranges, such as morning, afternoon and evening:

```
acl machine1 src 192.168.0.2
acl machine2 src 192.168.0.3
acl machine3 src 192.168.0.4
acl morning time 06:00-12:00
acl lunch time 12:00-14:00
acl afternoon time 14:00-17:30
acl evening time 17:30-06:00

http_access allow machine1 machine2
morning
http_access allow machine2 afternoon
http_access allow machine1 machine2
machine3 lunch evening
http_access deny all
```

This example allows web access in the morning to machines 1 and 2, all three at lunchtime and in the evening, but only machine 2 in the afternoon.

BLOCK AND KEY

Probably the most useful restriction you can impose is over which sites your users can visit. You can specify a destination domain to block, such as `youtube.com`:

```
acl no_youtube dstdomain .youtube.com
http_access deny no_youtube
```

This example will deny all access to any domain ending in `.youtube.com` but still allow all others. Use this if your users are blocking your internet connection by viewing YouTube videos all day. You can use regular expressions (regex) in the domain definitions too, using the `dstdom_regex` type:

```
acl no_triple_x dstdom_regex *xxx*
```

If you're going to block a large number of domains, you can point the ACL at a file containing a list of banned sites:

```
acl blocked_domains dstdomain "/etc/
squid/blocked_domains"
```

Note the use of the quotes to indicate the filename. In the file itself, simply place one domain name on each line. Combining domain blocking with a time-based ACL can give you automatic control over access to bandwidth-hungry sites such as YouTube and MySpace, restricting access to them to certain parts of the day. It may be that only certain users can't be trusted not to abuse their access. If those users are on Linux or UNIX machines, and are running the ident server on their installation, you can identify them and weed out their activities with the ident type:

```
ident_lookup on
acl bad_users ident bob alice
```

```
http_access deny bad_users no_youtube
morning afternoon
```

This example uses the 'morning' and 'afternoon' ACLs from the previous examples, and will keep users Bob and Alice away from YouTube during those times, but allow them on during lunch and after work.

You can also specify the inverse of an ACL using the bang (!) operator. Here's an example:

```
acl bad_users ident bob alice
http_access deny !bad_users no_
youtube morning afternoon
```

The bang before `bad_users` tells Squid to match all users except Bob and Alice. In this case, Bob and Alice are the only ones allowed to waste working hours using YouTube and MySpace. You can use the bang symbol to negate any ACL you choose.

ACLs are capable of expressing some complex allow and deny rules. To help you understand all the options and how to get the best from them, the developers of Squid have put together a very useful Wiki at www.deckle.co.za/squid-users-guide/Main_Page. There's also a reference guide to all Squid's configuration options at www.visolve.com/squid/squid24s1/contents.php.

TROUBLESHOOTING

One problem with using Squid is that some tech-savvy users might decide to change their browser settings to get direct access to whatever webpages they like. You can easily force them to use the proxy again by implementing a rule on your site's firewall that accepts only outgoing HTTP, HTTPS and FTP connections from the machine running Squid and discards all others.

Squid can help to speed up your web access by caching objects that appear on pages that users visit regularly but, like all caching proxies, there are limits to the amount it can help. The remote web server could be slow because it's oversubscribed or underpowered. In that case, unless the pages it serves consist of static objects such as plain images and HTML files, the bulk of the delay is probably spent waiting for the remote server to fulfil backend database requests. In this case, Squid won't significantly improve access times.

One other common problem is that you receive the following message when attempting to start Squid:

```
# /etc/init.d/squid start
/etc/init.d/squid: line 48: ulimit:
open files: cannot modify limit:
Operation not permitted
Starting WWW-proxy squid (/var/cache/
squid) - Could not create cache_dir!
Failed
```

This usually happens when attempting to start the proxy from an account other than root.

Finally, when creating ACLs, remember to start simply and combine them. If all else fails, remember to check your log files. Other than that, Squid is simple to set up and run. **CS**

Next month

VIRTUAL PRIVATE NETWORKS
Secure your network with a VPN